Remarks

The present amendment responds to the final Official Action dated November 3, 2003. The Official Action objected to the title of the invention. The Official Action objected to claims 1, 8, and 11. Claims 1, 2, 6-8, and 11 were rejected under 35 U.S.C. §102(b) based on Lee et al. U.S. Patent No. 4,763,242 ("Lee"). Dependent claims 3 and 4 were rejected under 35 U.S.C. §103(a) as being unpatentable over Lee in view of Dowling U. S. Patent No. 6,128,728 ("Dowling '728"). Dependent claim 5 was rejected under 35 U.S.C. 103(a) as being unpatentable over Lee in view of Dowling U.S. Patent No. 6,170,051 ("Dowling '051"). Claims 8-11 were rejected under 35 U.S.C. 103(a) as being unpatentable over Dahl et al. U.S. Patent No. 5,710,938 ("Dahl") in view of Lee. Dependent claims 12 and 13 were rejected under 35 U.S.C. 103(a) as being unpatentable over Dahl in view of Lee, in further view of Mirsky et al. U.S. Patent No. 5,915,123 ("Mirsky"). These grounds of rejection are addressed below.

Claim 2 has been canceled without prejudice. Claims 1, 6, 8, and 11 have been amended to be more clear and distinct. Claims 1 and 3-13 are presently pending.

Objection to the Title of the Invention

Although the Applicant believes that the previous title was both technically accurate and descriptive, the title of the invention has been modified. If the Examiner maintains this objection, the Applicant requests that the Examiner offer a suggested title or a call to further clarify the objection.

Amendment to the Specification of paragraph beginning at page 8, line 5

During review of the prosecution history of this application, it was noticed that the track change indication for the paragraph amendment was omitted. The paragraph has been amended with the appropriate track change indications.

Objections to claims 1, 8, and 11

Claim 1 was objected to because the phrase "an array one-by-one processor" was unclear. Claim 1 has been amended to replace the objected to phrase with the phrase "a merged sequence processor (SP) and processor element (PE) array processor" to reflect that the 1x1 array core can be viewed as a single processor containing two register contexts that share a common set of execution units. See the Specification, page 3, lines 1-10.

Claim 8 was objected to because of the informality of the phrase "additional register file files." The Examiner is thanked for his careful reading of the claims. Claim 8 has been amended in accordance with the Examiner's suggestion by removing the extra term "file."

Claim 11 was objected to for using an incorrect article before the term "array". Claim 11 has been amended in accordance with the Examiner's suggestion by replacing the phrase "a array" with the phrase "an array."

The Art Rejections

All of the art rejections hinge on the application of Lee or various combinations of Lee with Dowling '728, Dowling '051, Dahl, or Mirsky. As addressed in greater detail below, the relied upon art does not support the Official Action's reading of it and the rejections based

thereupon should be reconsidered and withdrawn. Further, the Applicant does not acquiesce in the analysis of the relied upon art made by the Official Action and respectfully traverses the Official Action's analysis underlying its rejections.

Lee describes a system which includes a main processor and an assist, or hardware which extends the main processor's capabilities by providing support for additional extension instructions which are not part of the main processor's basic instruction set. This assist may be in the form of a coprocessor or a special function unit. When an instruction is fetched from memory, a field in the instruction is decoded to determine whether the instruction is a basic instruction or an extension instruction. If the instruction is a basic instruction, it is executed by the main processor. If the instruction is an extension instruction, the field is further decoded to determine which assist to route the instruction to for execution. Lee makes no determination of which register file to use during the execution of the instruction. The choice of register file to use is fixed to the choice of which processor executes the instruction. The Official Action admits this fixed relationship when it states "If a COP instruction is encountered then the COP register file will be used while the COP is executing the instruction. On the other hand, if the main processor is executing the instruction, then its register file 119 will be accessed." See Fig. 3, component 339.

In contrast to Lee, the present invention provides techniques for efficient context switching between software tasks executing in an array processing environment. In one aspect of the present invention, an array controller sequence processor (SP) is merged with a processing element (PE) in order to share execution units between the SP and PE. Consequently, in the

merged SP/PE, a single set of execution units are coupled with two independent register files. To make efficient use of the SP and PE resources, a bit in the instruction format, the SP/PE bit, differentiates SP instructions from PE instructions. SP instructions include instructions which are executed sequentially while PE instructions are instructions which can be executed in parallel with other instructions or another PE.

Multiple register contexts may be obtained by controlling how the SP/PE bit in the instruction format is used in conjunction with a context switch bit (CSB) for the context selection of the PE register file or the SP register file. The selection between the PE register file or the SP register file for use with a sequential instruction advantageously provides reconfiguration of PE and SP resources to address software that favors sequential instruction processing. See amended claim 1, for example, which recites:

> 1. Apparatus for providing efficient context switching between software tasks in a merged sequence processor (SP) and processor element (PE) processor environment, each software task comprising a plurality of instructions, the SP by one PE array processor environment configurable to be in a first array configuration or a second array configuration, the apparatus comprising:
>
> a first set of registers stored in the SP register file;
>
> a second set of registers stored in the PE register file;
>
> a sequence processor/processing element (SP/PE) selection bit in an instruction; and
>
> a context select bit (CSB) in a processor state register, a specific instruction out of the plurality of instructions setting the CSB, the CSB in conjunction with the SP/PE selection bit selecting a context of a first software task utilizing the first array configuration or a context of a second software task utilizing a second array configuration, the first array configuration including at least one register from the second set of registers to execute sequential instructions, the second array configuration including at least one register from the first set of registers to execute sequential instructions. (emphasis added)

See also amended claim 8 which recites that "reconfigures the array by selecting a first context in which the array is configured in a first configuration which provides sequential instructions to utilize one of the plurality of additional register files or a second context in which the array is configured in a second configuration utilizing the first set of registers for sequential instructions" and amended claim 11 which recites "configuring the array processing to have either a first configuration or a second configuration depending upon the context, the first configuration including at least one register file of the plurality of PE register files for a sequential instruction, the second configuration including the SP register file for a sequential instruction."

Lee does not teach and does not render obvious such techniques for sharing register files. In fact, Lee teaches away from sharing of register files by always using the register file within the main processor or an assist, whichever executes the instruction. Thus, Lee does not provide sharing of register files between basic instructions and extension instructions. Lee merely allocates basic instructions to be executed on the main processor and extension instructions to be executed on the assist.

The Official Action asserts that the assist bit field described at col. 4, lines 66-68 of Lee is the same as the context select bit (CSB) stored in a processor state register. At the cited portion of Lee, the assist bit field is described as imbedded in the instruction. This field only indicates which hardware assist should be used for executing the instruction, and is not a context select bit stored in a processor state register, as presently claimed.

The Official Action in the Response to Arguments section further asserts that "the instruction itself is inherently stored within an instruction register (processor state register)." The

Official Action cites Hamacher et al., Computer Organization, 4th Edition, 1996 ("Hamacher"), p. 7 and 112-113 for support of this inherency statement. The Official Action misinterprets the role of an instruction register (IR) as described by Hamacher. At page 7, line 28 of Hamacher, an instruction register is defined as holding "the instruction that is currently being executed. Its output is available to the control circuit, which generate the timing signals that control the various processing elements involved in executing the instruction." As each instruction is executed, the IR's contents change to hold a new current instruction. A processor state register(PSR) which is well known to one skilled in the art is different in that it contains data that persists between instructions. The present invention modifies the PSR by a specific instruction. Claim 1, as presently amended, recites "a specific instruction out of the plurality of instructions setting the CSB." See also claims 8 and 11, for example.

Regarding claims 8-11, Dahl does not cure the failings of Lee as a reference. Dahl teaches an array system in which the array is partitioned into multiple sub-arrays which operate independently of each other. In Fig. 1 of Dahl, the array 11 is configured by a control processor 20. The control processor 20 does not have direct access to register files of any processor node in array 11 as it communicates with the processor nodes and network nodes via bi-serial control channels for the purpose of partitioning the array. See Dahl, col. 4, lines 3-6 and lines 11-16. As indicated by the Official Action, Dahl does not teach apparatus for providing efficient context switching between tasks. Applicants agree. As addressed above, Lee also fails to teach context switching between tasks as presently claimed. Thus, the proposed combination of these two items does not teach and does not make obvious the present claims.

Further, the Official Action admits the lack of sharing between register files of different processing elements when it states that "Dahl could assign different tasks to different processing elements, where each task deals with its own register file contents." (emphasis added) Thus, the proposed combination of Dahl and Lee does not teach and does not make obvious the sharing of register files between the SP and the PE as presently claimed.

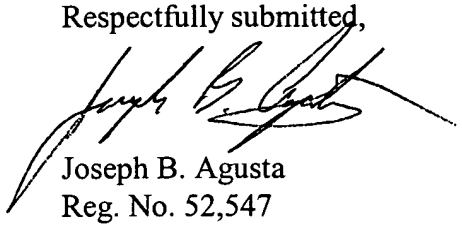
Regarding dependent claims 3-5, 12 and 13, the other relied upon references do not cure the failings of Lee as a reference. Dowling '728 teaches a system which utilizes shadow register sets and shadow windows to transfer data between registers and memory in burst and cycle steal modes. Dowling '051 describes a processor which allows one program to execute in a cycle steal mode to make use of the inefficiencies of another program. Mirsky teaches techniques for providing local control of processing elements in a network of multiple context processing elements. These references do not teach and do not render obvious the presently claimed techniques for context switching.

In summary, the relied upon art does not indicate a recognition of the problems addressed by the present invention. Further, the relied upon art does not teach or suggest an apparatus which would solve the problems of context switching on array processors addressed by the present invention in the manner solved by the present invention. The claims as presently amended are not taught, are not inherent, and are not obvious in light of the relied upon art.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,



Joseph B. Agusta
Reg. No. 52,547
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919) 806-1600